AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listing, of claims in the application:

Claim 1 (Currently amended): A method for verifying the validity of an encrypted code generated in base L, the method comprising ~~the steps of~~:

obtaining an encrypted code from a user, that when decrypted is determinable to indicate a value~~fashioned as a~~ in base L, the encrypted code obtained by appending a third string, which is an output of applying an encryption algorithm employing a second secret code to a second string composed of an n-bit raw number and an m-bit validation number, wherein m is at least 16, the m-bit validation number generated by hashing, with a hash function, a first string, the first string composed of the n-bit raw number and the first secret code;
~~string derived from an n-bit raw number by producing a first string through application of a one-way hash function to the n-bit raw number of a one-way hash function to the n-bit raw number with a first secret key,~~

~~designating an m-bit portion of the first string as an m-bit validation number,~~

~~producing a second string through combination of the m-bit validation number and the n-bit raw number, producing a third string through application of an encryption algorithm to the second string with a second secret code, and converting the third string to the base L string~~;

converting the encrypted code in base L ~~string~~ to a base 2 string;

decrypting the base 2 string using the second secret code to generate a recovered second string;

hashing, with the first secret code, an n-bit portion of the recovered second string concatenated with the first secret code to generate a second m-bit validation number;

comparing the remaining m-bits of the recovered second string with the second m-bit validation number to verify the validity of the encrypted code; and

if valid then crediting the user with the value indicated by the decrypted code. ~~and~~

~~verifying the validity of the encrypted code by processing the decrypted base 2 string.~~

Claim 2 (Previously presented): The method of claim 1, wherein the encryption algorithm is a DES3 encryption algorithm.

Claim 3 (Currently amended): ~~The method of claim 1, wherein n=32, m=16, and L=29.~~A method for verifying the validity of an encrypted code generated in base L, the method comprising ~~the steps of:~~

obtaining an encrypted code from a user, that when decrypted is determinable to indicate a value-in base L, the encrypted code obtained by appending a third string, which is an output of applying an encryption algorithm employing a second secret code to a second string composed of an n-bit raw number and an m-bit validation number, the m-bit validation number generated by hashing, with a hash function, a first string with a first secret code, the first string composed of the n-bit raw number and the first secret code;

converting the encrypted code to a base 2 string;

decrypting the base 2 string using the second secret code to generate a recovered second string;

hashing, with the first secret code, an n-bit portion of the recovered second string concatenated with the first secret code to generate a second m-bit validation number;

comparing the remaining m-bits of the recovered second string with the second m-bit validation number to verify the validity of the encrypted code; and

if valid then crediting the user with the value indicated by the decrypted code, wherein n=32 and m=16.

Claim 4 (Currently amended): The method of claim 1, wherein the ~~one-way~~ hash function is MD5.

Claim 5 (Cancelled)

Claim 6 (Previously presented): The method of claim 1, wherein the m-bit validation number is the m most significant bit (MSB) portion of the second string.

Claims 7-11 (Cancelled)

Claim 12 (Currently amended): A method for awarding incentive points to a user, comprising ~~the steps of~~:receiving on-line from [[a]] the user a code generated with encrypted information and obtained by the user [[off]] offline;

verifying the validity of the code by processing the encrypted information; and

awarding incentive points to the user if the code is valid, wherein verifying includes converting a base L string of the code to produce a first test code, decrypting the first test code using a second secret key to obtain a second test code, the second test code comprising an m-bit validation number and an n-bit number, applying a one-way hash function to the n-bit number with a first secret key to produce a hash output, comparing an m-bit portion of the hash output to the m-bit validation code to determine if the code is valid,

wherein m is at least 16.

Claim 13 (Previously presented): The method of claim 12, wherein the code is generated by:

providing an n-bit raw number;

generating a first string through application of a one-way hash function to the n-bit raw number with a first secret key;

designating an m-bit portion of the first string as an m-bit validation number;

generating a second string through combination of the m-bit validation number and the n-bit raw number;

generating a third string through application of a DES encryption algorithm to the second string with a second secret key; and

producing the code with the encrypted information through conversion of the third string to a base L string.

14. (Currently amended) The method of claim 12, wherein the step of verifying includes:

generating a first test code by converting the base L string of the code to a base 2 string;

generating a second test code by decrypting the first test code with the second secret key using a reverse DES3 encryption algorithm;

generating a third[[st]] test code by applying the one-way hash algorithm to the second test code; and

determining the validity of the code by comparing a designated m-bit portion of the second test code to a designated m-bit portion of the third test code,

wherein n=32 and m is at least 16.

Claims 15-16 (Cancelled)

Claim 17 (Currently amended): An offline-online points system, comprising:

a main server configured with an interface for receiving a code from a user, wherein the code is obtainable by the user off-line and is associated with N points, wherein each point,

characterized as a purchase or attention incentive point, is redeemable and maintainable in an account for the user; and

a code server configured for maintaining valid codes and verifying~~, against the valid codes,~~ the validity of the code received from the user, wherein the account has a balance of points capable of growing by a predetermined number of points if the code is valid, the verifying proceeding by converting a base L string to produce a third string, decrypting the third string using a second secret key to obtain a second string, the second string comprising an m-bit validation number and an n-bit number, applying a one-way hash function to the n-bit number with a first secret key to produce a hash output, and comparing an m-bit portion of the hash output to the m-bit validation code to determine if the code is valid,

wherein m is at least 16.

Claims 18 – 20 (Cancelled)

Claim 21 (Currently amended): ~~The method of claim 18~~ A method for awarding incentive points to a user, comprising the steps of:

receiving on-line from a user a code generated with encrypted information and obtained by the user off-line;

verifying the validity of the code by processing the encrypted information; and

awarding incentive points to the user if the code is valid,

wherein the code is generated by:

providing a number portion,

deriving a validation portion from the number portion,

appending the validation portion to the number portion to form a string,

encrypting the string, and

deriving the code from the encrypted string by converting the encrypted string to base L string,

wherein the string is 48-bits long and the number portion is 32-bits long.

Claim 22 (Currently amended) The method of claim 12, wherein the code is generated by:

providing a number portion, Sl_{INT}, from a first string, S1

arranging a first secret key, K1, next to the number portion, Sl_{INT}, from S1, to form a second string, S2,

applying a hash function to S2 to produce a third string, S3,

extracting a validation portion, $S1_{VAL}$, from S3 and arranging $S1_{VAL}$ next to Sl_{INT} in S1 ($S1 = S1_{VAL},+ Sl_{INT}$),

encrypting S1 using a second secret key, K2, to form a fourth string, S4, and

deriving the code by converting S4 to a base L fixed-length code string,

wherein S3 is at least 16 bits long.

Claim 23 (Previously presented): The method of claim 22, wherein the first and second secret keys, K1 and K2, are 128-bits long and the encryption includes DES3 encryption algorithm.

Claim 24 (Previously presented): The method of claim 22, wherein the hash function includes MD5, a one-way hash algorithm.

Claim 25 (Currently amended): ~~The method of claim 22~~A computer-enabled method for awarding incentive points to a user, comprising:

receiving on-line from the user a code generated with encrypted information and obtained by the user offline;

verifying the validity of the code by processing the encrypted information; and

awarding incentive points to the user if the code is valid, wherein the code is generated by:

providing a number portion, Sl_{INT}, from a first string, S1

arranging a first secret key, K1, next to the number portion, Sl_{INT}, from S1, to form a second string, S2,

applying a hash function to S2 to produce a third string, S3,

extracting a validation portion, $S1_{VAL}$, from S3 and arranging $S1_{VAL}$ next to Sl_{INT} in S1 ($S1 = S1_{VAL},+ Sl_{INT}$),

encrypting S1 using a second secret key, K2, to form a fourth string, S4, and

deriving the code by converting S4 to a base L fixed-length code string,

wherein S1 is 48-bits long and the number portion, Sl_{INT}, is 32-bits long.

Claims 26-28 (Cancelled)

Claim 29 (Currently amended): A method for offline-online management of incentive points, comprising:

receiving a code, generated by providing a number portion, deriving a validation portion from the number portion, appending the validation portion to the number portion to form a string, encrypting the string, and deriving the code from the encrypted string by converting the encrypted string to a base L string, the code obtained off-line and received on-line; [[and]]

~~processing the code.~~

submitting the code to a server that has valid codes, wherein the code is associated with N points maintained by the server in a user account, wherein each point, characterized as a purchase or attention incentive point, is redeemable; and

verifying the code against the valid codes to determine if it is valid, wherein if the code is valid, a predetermined number of points are added to the user account,

wherein the validation portion is at least 16 bits long.

Claims 30-31 (Cancelled)

Claim 32 (Currently amended): [[A]] The method ~~as in~~ of claim 29, wherein the string is at least 48-bits long and the number portion is 32-bits long. ~~L is the number of characters in the alphabet.~~

Claim 33 (Currently amended): A method ~~as in claim 29~~ for offline-online management of incentive points, comprising:

receiving a code, generated by providing a number portion, deriving a validation portion from the number portion, appending the validation portion to the number portion to form a string, encrypting the string, and deriving the code from the encrypted string by converting the encrypted string to base L string, the code obtained off-line and received on-line; and

submitting the code to a server that has valid codes, wherein the code is associated with N points maintained by the server in a user account, wherein each point, characterized as a purchase or attention incentive point, is redeemable; and

verifying the code against the valid codes to determine if it is valid, wherein if the code is valid, a predetermined number of points are added to the user account,

wherein the string is 48-bits long and the number portion is 32-bits long.

Claims 34-36 (Cancelled)

Claim 37 (Currently amended): ~~A method as in claim 34~~A method for generating a code that corresponds to incentive points, comprising:

providing a number portion, Sl_{INT};

arranging a first secret key, K1, next to Sl_{INT}, to form a second string, S2,

applying a hash function to S2 to produce a third string, S3, extracting a validation portion, $S1_{VAL}$, from S3 and arranging $S1_{VAL}$, next to Sl_{INT} to produce] S1 ($S1 = Sl_{VAL} + Sl_{INT}$),

encrypting S1 using a second secret key, K2, to form a fourth string, S4, and

deriving the code by converting S4 to a base L fixed-length code string; and

fixing the code onto a medium such that the code is obtainable from the medium off-line,

wherein S1 is 48-bits long and the number portion, Sl_{INT}, is 32-bits long.

Claim 38 (Currently amended): A method as in claim [[30]] 29 wherein the step of verifying the ~~submitted~~ code includes,

converting the ~~submitted~~ code from a base L string into a base 2 string, $S4_{BASE2}$,

decrypting $S4_{BASE2}$ using a second secret key, K2, to form a decrypted first string, S1',

providing a number portion from S'

arranging a first secret key, K1, next to the number portion from S1' to form a second string, S2',

applying a hash function to S2' to form a third string S3',

extracting a validation portion from S3' and a validation portion from S1', and

determining if the code is valid by comparing the validation portion from S3' with the validation portion from S1',

wherein S3' is at least 16 bits long.

Claim 39 (Currently amended): ~~A method as in claim 38, wherein S3' and S1 are each 48 bits long and the secret keys, K1 and K2 are 128 bits long.~~ A method as in claim 38, wherein S1' is 48-bits long and the number portion is 32-bits long.

Claim 40 (Cancelled)